

Remarks/Arguments

Reconsideration of this application is requested.

Extension of Time

A request for a one month extension of time for response to the Office Action mailed January 7, 2005 is enclosed. The extended period for response expires on May 9, 2005.

Claim Status

Claims 1-109 were presented and remain pending. Claims 1, 6, 10, 18, 26, 33, 49, 51, 53, 68, 77, 90, 92, 94, and 108 are amended.

Title

The title is objected to as non-descriptive. In response, the title is amended to "System and Method for Extracting Software Inter-Component Dependencies for Re-Use, Re-Engineering and Comprehension" to be clearly indicative of the invention to which the claims are directed.

Claim Rejections – 35 USC 101

Claims 1-89 and 108-109 are rejected under 35 USC 101 as directed to non-statutory subject matter. In response, claims 1, 6, 10, 18, 26, 33, 49, 51, 53, 68, 77 and 108 are amended as suggested in the Action.

Claim Rejections – 35 USC 102(b)

Claims 1 and 3-109 are rejected under 35 USC §102(b) as anticipated by Rational Rose, version 4.0, released November 1996 ("Rational Rose").

The present invention is directed to algorithmic logic and the heuristics of "cleanly" extracting visual components from source code to enable and promote the reuse of such components. A software architecture relationship is extracted without requiring the developer to follow a predefined set of rules, coding styles or guidelines. This technique is applicable to source code where the original developer is either unknown, unavailable, or has simply forgotten the architectural blueprint of his original design, and attempts to recreate the architectural view of the original

designer by mining and analyzing the design and applying a unique inference mechanism.

This technique is orthogonal to, but not predicated on, concepts of object oriented design. The present invention does not require or presume any knowledge of object oriented design or analysis, and is applicable to software designs that are developed in programming languages that do not natively support any form of object oriented design or development paradigms. A multitude of different design and presentation styles of the architecture may be specified based on user choices and configurations.

These principles of the invention are reflected in the claims. Claim 1, for example, is amended to require generation of a plurality of *language independent format objects* from a plurality of *language dependent format objects*. The language independent format objects represent discrete components that are structurally related into an information model according to a software architecture. Support for these features is found throughout the specification. See, for example, paragraph [0088].

The remaining independent claims contain, or are amended to contain, similar limitations. Claim 6, as filed, recites extraction of program fragments from a body of source code and creation of a plurality of language dependent format objects, and conversion of the language dependent format objects into language independent format objects. Claims 10, 18, 26, 33, 49, 51, 53, 68, 77, 90, 92, 94, and 108 contain or are amended to contain analogous limitations.

Rational Rose fails to meet these limitations. Rational Rose requires a designer to map artifacts in the original design, such as classes, etc., to OMT or UML artifacts, such as components. Rational Rose only allows such associations between C++ programs and UML or OMT artifacts and does not allow other kinds of associations between C++ programs and other notations that are frequently used to illustrate software architectures. Rational Rose does not describe how associations can be created for programs that do not have any language provided

support for object oriented design or other forms of architecture specification. In other words, Rational Rose describes only generation of language dependent objects. It does not teach, as required by all claims of the present application, generation of a plurality of *language independent format objects* from a plurality of *language dependent format objects*.

At Chapter 2, page 28, for example, Rational Rose describes a C++ generator that uses specifications and code generation properties of components in a current model to produce C++ source code, member functions, and source code files. Thus, Rational Rose produces source code from components in a current model in a language dependent format only.

Because Rational Rose does not disclose each and every element of claims 1 and 3-109, it cannot anticipate those claims.

Claim Rejections – 35 USC 103(a)

Claim 2 is rejected under 35 USC 103(a) as obvious over Rational Rose in view of “XML Makes Object Models More Useful”, Bruce Klein, Information Week, June 28, 1999. This ancillary reference does not overcome the deficiencies of Rational Rose, discussed above. Claim 2 is therefore allowable for the same reason discussed with reference to claim 1 and 3-109.

Conclusion

This application is now believed to be in form for allowance. The examiner is invited to telephone the undersigned to resolve any issues that remain after entry of this amendment.

Appl. No. 09/865,300
Amdt. Dated May 6, 2005
Reply to Office Action of January 7, 2005

Attorney Docket No. 88537.0008
Customer No. 26021

Any fees due with this response may be charged to our Deposit Account No.
50-1314.

Respectfully submitted,
HOGAN & HARTSON L.L.P.

Date: May 6, 2005

By: 

Troy M. Schmelzer
Registration No. 36,667
Attorney for Applicant(s)

500 South Grand Avenue, Suite 1900
Los Angeles, California 90071
Phone: 213-337-6700
Fax: 213-337-6701